Concatenation of Short Constraint Length Convolutional Codes

D. Erickson
Communications Systems Research Section

Several methods of decoding a concatenated pair of $K = 6$, $V = 2$ convolutional codes are investigated. It was found that none of the methods provides performance which is suitable for space channel application.

I. Convolutional Codes

An encoder for a convolutional code consists of a K-bit shift register and V multi-input binary adders, each with some combination of the bits of the shift register as inputs. As each input bit is shifted into the register, the V outputs from the adders are transmitted. The rate of such a code is $1/V$, and K is called its constraint length. Such a code will be referred to as a K by V code. The code may be described by specifying its V by K tap matrix $A = (a_{ij})$ where $a_{ij} = 1$ if the jth bit of the shift register is connected to the ith binary adder $a_{ij} = 0$ otherwise. If the input to the encoder is $\cdots, \phi_t, \phi_{t+1}, \phi_{t+2}, \cdots$ the output will be $\cdots, (b_{t,1}, b_{t,2}, \cdots, b_{t,V}), (b_{t+1,1}, b_{t+1,2}, \cdots, b_{t+1,V}), \cdots$ where $(b_{t1}, \cdots, b_{tV})^t = A(\phi_t, \phi_{t-1}, \cdots, \phi_{t-K+1})^t$. The tap matrix is often abbreviated by representing its columns in hexadecimal notation if $V \leq 4$. This is accomplished for the jth column by representing $a_{V,j} + 2a_{V-1,j} + \cdots + 2^{V-1} a_{1,j}$ as a hexadecimal digit $0, 1, \cdots, F$ (Fig. 1). The signal transmitted is usually $c = \cdots, (c_{t,1}, \cdots, c_{t,V}), \cdots$ where $c_{ij} = (-1)^{b_{ij}}$. Thus we transmit ± 1 instead of 0,1. Notice in Fig. 1 how the response of the encoder to an impulse input is the columns of the tap matrix.

II. Viterbi Decoding

In the Viterbi algorithm for decoding a convolutional code, one views the encoder as a finite state machine with 2^{K-1} possible states corresponding to the contents of the first $K - 1$ bits of the shift register (Ref. 1). As each new bit is shifted in, the encoder generates V output bits and changes states. The encoder 313 pictured in Fig. 1 has the four states 00, 01, 10, and 11. If it is in state 10 and receives a 0 as the next input, it outputs 1, 0 and changes to state 01. If the message $\mathbf{y} = \cdots, y_i, y_{i+1}, \cdots$ is encoded into $\mathbf{x} = \cdots, (x_{i1}, \cdots, x_{iV}), \cdots$ and the channel error is $\mathbf{e} = \cdots, (e_{i1}, \cdots, e_{iV}), \cdots$ then the received signal will be $\mathbf{r} = \mathbf{x} + \mathbf{e}$. The decoder wishes to find $\mathbf{y}$ given $\mathbf{r}$. To do this, he must find the message $\mathbf{\Phi}$ which encodes into the signal $\mathbf{c} = \mathbf{c}(\mathbf{\Phi})$ such that $\mathbf{e} = \mathbf{e}(\mathbf{\Phi}) = \mathbf{r} - \mathbf{c}$ is minimized. We minimize $\mathbf{e}$ by minimizing the sum

$$\sum_{t=-\infty}^{\infty} \sum_{j=1}^{V} e_{ij}^2 - \bar{e}_{ij}^2$$

where $e_{ij} = r_{ij} - c_{ij}$ and $\bar{e}_{ij} = r_{ij} - (-c_{ij}) = r_{ij} + c_{ij}$. This sum reduces to

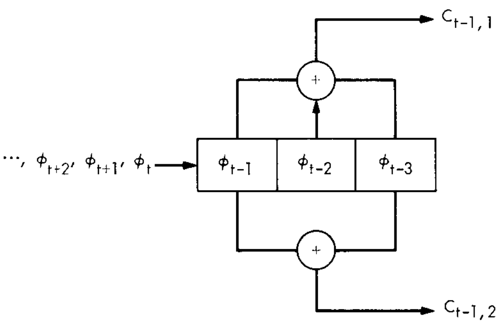
$$\sum_{t=-\infty}^{\infty} \sum_{j=1}^{V} -4r_{ij}c_{ij}$$

If we define the metric of $\mathbf{\Phi}$ at time t

$$M(\mathbf{\Phi}, t) = \sum_{t=-\infty}^{t} \sum_{j=1}^{V} r_{ij}c_{ij}$$

then our task is to maximize $M(\mathbf{\Phi}, \infty)$.

A 3 BY 2 CONVOLUTIONAL CODE



TAP MATRIX $A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

ABBREVIATED REPRESENTATION 323

DECIMAL-TO-HEX CONVERSIONS

DECIMAL VALUE	0	1 · · ·	9	10	11	12	13	14	15
HEX DIGIT	0	1 · · ·	9	A	B	C	D	E	F

IMPULSE RESPONSE - ENCODING OF ···0 0 0 1 0 0 0···

TIME	SHIFT REGISTER	OUTPUT $c_{t,1}$	OUTPUT $c_{t,2}$	STATE	SIGNAL
t_{0-1}	0 0 0	0	0	0 0	1, 1
t_0	1 0 0	1	1	1 0	-1, -1
t_{0+1}	0 1 0	1	0	0 1	-1, 1
t_{0+2}	0 0 1	1	1	0 0	-1, -1
t_{0+3}	0 0 0	0	0	0 0	1, 1

Fig. 1. A 3 by 2 convolutional code

For each message Φ, we also know $S(\Phi, t)$ the state in which the encoder will be at step t. For any states, let the survivor $\Phi(s,t) = \cdots, \phi_{t-1}, \phi_t$ be the message Φ up to step t which maximizes $M(\Phi, t)$ given $S(\Phi, t) = s$. Let $m(s,t)$ be its metric. If we know for each state s, $\Phi(s, t-1)$ and $m(s, t-1)$, we can find $\Phi(s,t)$ and $m(s,t)$ as follows: Each state s can only result from one of two previous states s_0 and s_1. That is, the state 01 can result from 10 or 11. If we know the present state and the previous state, then we know the contents of the shift register and, hence, the output for the step.

Knowing the step output gives us the change in metric m_0 or m_1. We compare $m_0 + m(s_0, t-1)$ and $m_1 + m(s_1, t-1)$ and the larger is $m(s,t)$. Its corresponding survivor Φ becomes $\Phi(s,t)$, the new survivor for s with a 1 or 0 appended, depending on whether s begins with a 1 or 0. These survivors tend to converge after 4 or 5 constraint lengths (Ref. 2, pp. 61–64) so that if $\Phi(s,t) = \cdots, \phi_{t-1}, \phi_t$ and $\Phi(s',t) = \cdots, \phi'_{t-1}, \phi'_t$ then ϕ_{t-5k} usually equals ϕ'_{t-5k}. Thus the survivors need only be saved to a finite depth.

The storage needed is proportional to $K \cdot 2^{K-1}$. Although long constraint length codes perform better, they require so much storage and time or circuitry to decode that they are impractical.

One method of improving performance is careful evaluation of the received signal. If the received signal is hard-limited (resolved into $+1$ or -1) performance is 2 dB worse than if it is resolved to 4-bit accuracy $-^{15}\!/_{16}$, $-^{13}\!/_{16}$, $\cdots$, $^{13}\!/_{16}$, $^{15}\!/_{16}$ (Fig. 2).

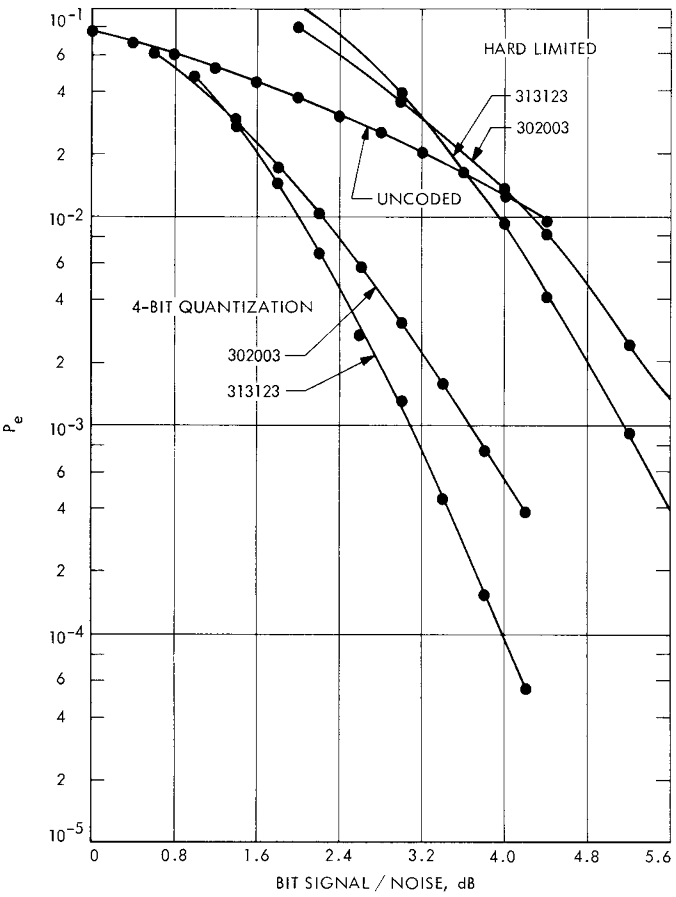


Fig. 2. Performance of 6 by 2 codes

III. Concatenation

Another possible method of improving performance without increasing complexity and storage requirements is to concatenate codes. In encoding, concatenation is accomplished by using the output from one encoder as input to a second encoder (Fig. 3). One way to find the characteristics of the concatenated code is to find its response to an impulse input, $\cdots, 0, 0, 0, 1, 0, 0, \cdots$. If the first encoder is a K_1 by V_1 code with tap matrix $T = (t_{ij})$, the output from it will be $\cdots$, $(0, \cdots, 0)$, $(t_{11}, t_{21}, \cdots, t_{V_1})$, $\cdots$, $(t_{1K_1}, \cdots, t_{V_1K_1})$, $(0, \cdots, 0), \cdots$. The output of the pair will be the response of the second encoder to this input. If the second code is K_2 by V_2, for each of the t_{ij}'s it will have V_2 outputs. Thus for each input bit, the pair will have $V_1 \cdot V_2$ outputs. The effect of the impulse will be felt by the pair until it has passed through the first encoder K_1 steps, and $t_{V_1K_1}$ has passed through the second encoder another $(K_2 - 1)/V_1$ steps. Rounded up, this gives a $K_1 + [(K_2 - 2)/V_1] + 1$ by V_1V_2 code. ($[x]$ is the greatest integer $\leqq x$.) Thus two 6 by 2 codes can be concatenated to give a 9 by 4 code. If this code could be decoded by two 6 by 2 decoders then the memory requirement would be cut by a factor of more than 4.

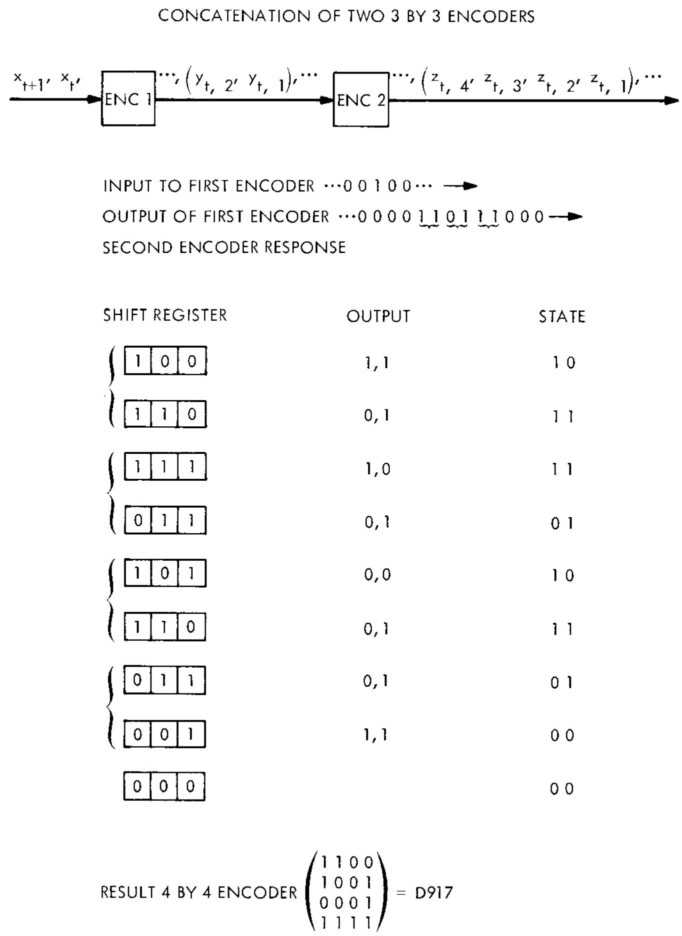


Fig. 3. Concatenation of two 3 by 2 encoders

Two questions arise:

(1) What form of output from the first decoder is best to use as input to the second decoder?

(2) Can concatenated decoding perform as well as or better than a single decoder of similar complexity?

To pursue these questions, a pair of 6 by 2 Viterbi decoders were simulated on an XDS 930 computer. The program used is an extension of the one described by J. Layland in Ref. 2, pp. 64–66.

IV. Methods of Linking

Figure 4 contains a block diagram for the three methods of linking described here: direct, differential, and democratic linkage.

Output from the inner decoder, which corresponds to the second encoder, is usually taken to be the oldest bit of the survivor of the most likely state (the s with $m(s,t)$ largest). If this output is ϕ_t (0 or 1), we can use $(-1)^{\phi_t}$ as input to the outer decoder. We will call this *direct linkage.*

Direct linkage has the disadvantage of providing only hard-limited inputs to the outer decoder which carry no judgments of confidence from the inner decoder. Just as 4-bit quantization improved the performance of a single decoder, we would hope that weighting the inner decoder's decisions would improve performance of the pair.

If the output from the inner decoder is $\boldsymbol{\Phi}$ we can calculate the metric of $\boldsymbol{\Phi}$,

$$M(\boldsymbol{\Phi}, t) = \sum_{i=-\infty}^{t} \sum_{j=1}^{V} r_{ij} c_{ij}$$

We can also calculate the metric of $\boldsymbol{\Phi}'$, the sequence differing from $\boldsymbol{\Phi}$ in the lth place. Then $M(\boldsymbol{\Phi}, \infty) - M(\boldsymbol{\Phi}', \infty)$ is a measure of the confidence in the lth bit of $\boldsymbol{\Phi}$. When $(-1)^{\phi_l}[M(\boldsymbol{\Phi}, \infty) - M(\boldsymbol{\Phi}', \infty)]$ is used as input to the second decoder, this is called *differential linkage.* Note that if $\mathbf{c}$ is the signal resulting from message $\boldsymbol{\Phi}$ and $\mathbf{c}'$ is the signal generated by Φ', then $\mathbf{c}$ and $\mathbf{c}'$ differ in only $K_2 \cdot V_2$ places at most. Thus

$$M(\Phi, \infty) - M(\Phi', \infty) = \sum_{i=l}^{l+K_2} \sum_{j=1}^{V_2} r_{ij}(c_{ij} - c'_{ij})$$

$$= \sum_{i=l}^{l+K_2} \sum_{j=1}^{V_2} r_{ij}c_{ij}(1 - c_{ij}\,c'_{ij})$$

$$= \sum_{i=l}^{l+K_2} \sum_{j=1}^{V_2} r_{ij}c_{ij}2t_{j,i-l}$$

where $T = (t_{ij})$ is the tap matrix of the outer code.

In the implementation described by the block diagram in Fig. 4, the received signal $\mathbf{r}$ is saved in a shift register. The hard output Φ from the inner decoder is encoded by a copy of the second encoder (marked ENC 2) and the resulting $\mathbf{c}$ is multiplied by $\mathbf{r}$ with the appropriate alignment so that the products $r_{ij}c_{ij}$ are formed. The linear combination of these corresponding to the tap matrix is then used as a weight for the hard output.

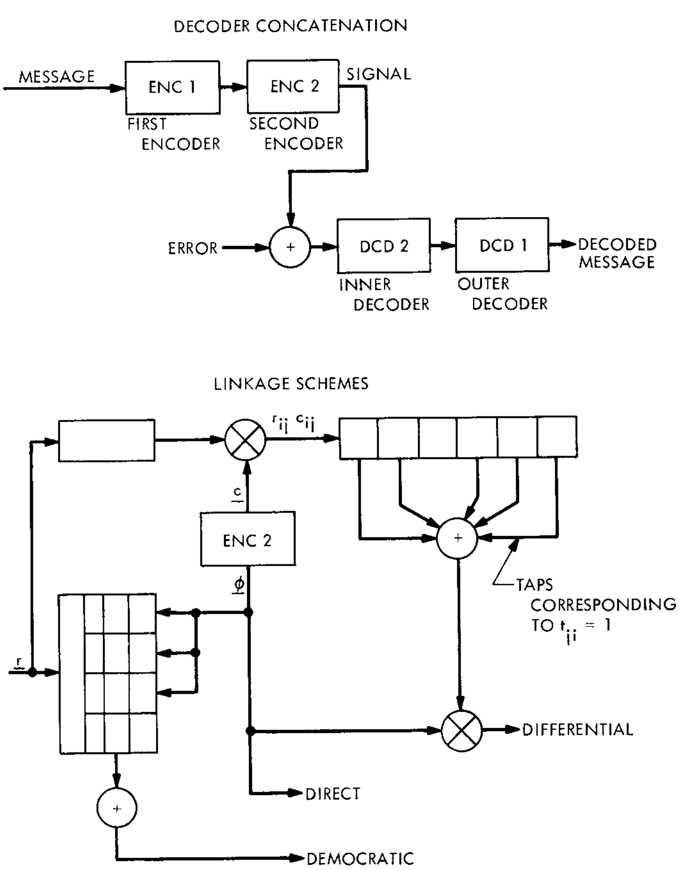


Fig. 4. Decoder concatenation and linkage schemes

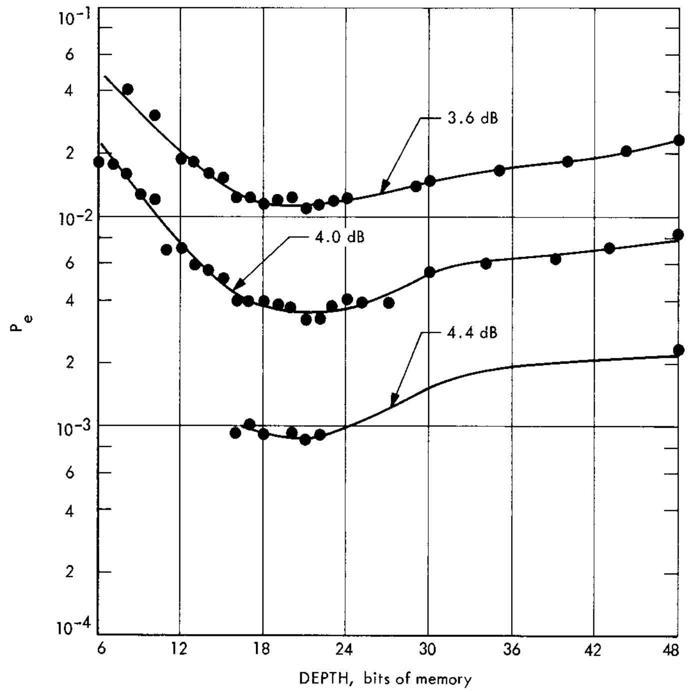


Fig. 5. Performance of democratic linkage versus depth

Another method of linkage is to let the bits of the survivors of all the states vote. Let $S_d(s)$ be the dth bit of the survivor of state s of the first decoder. When second decoder input is

$$\sum_s (-1)^{S_d(s)}$$

this is called *democratic linkage at depth d.* When the depth is too great, the survivors have converged and this is no different from using direct linkage. When d is too small, the probability of error is large. Thus, there is an optimal depth which may vary with the signal-to-noise ratio (Fig. 5).

Democratic linkage at this depth performs better than differential or direct linkage (Fig. 6).

V. Simulation Results

Results indicate that a pair of 6 by 2 decoders could not compete with a single 7 by 3 decoder except at very high signal-to-noise ratios which are not encountered in the planetary program. In fact, for the three methods of linkage tried, the pair did not perform as well as a single 6 by 2 decoder at the same bit signal-to-noise ratio. When the pair is working at noise ratios of interest e.g., $ST_b/N_0 = 3$ dB, the first decoder is operating at half this

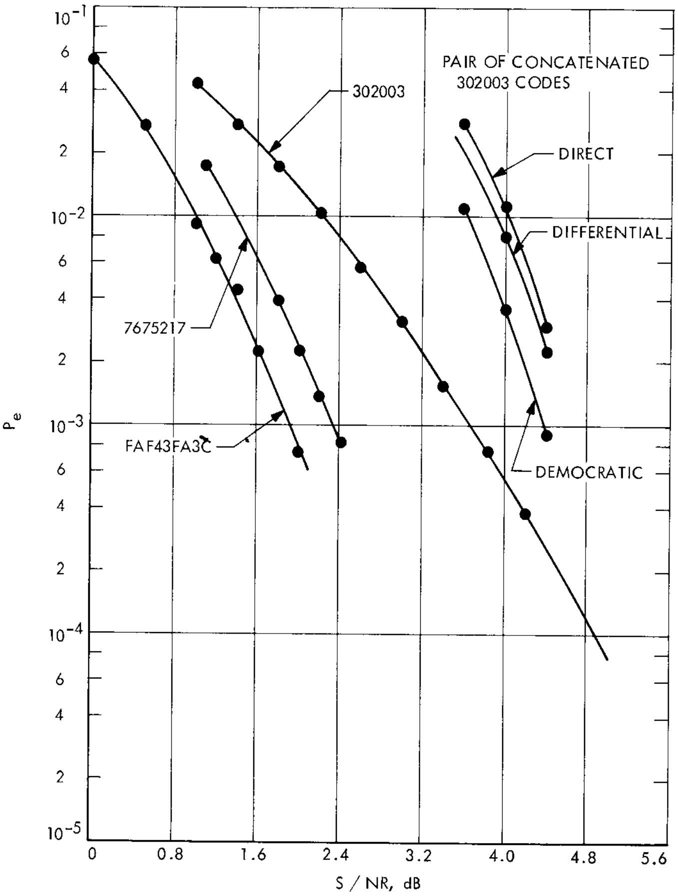


Fig. 6. Performance of concatenated code

power or 0 dB. At this power, it does not perform well enough to regain the lost 3 dB. Two 6×2 codes were studied. One (313123) performs better at higher signal-to-noise ratios, while the other (302003) performs better at low signal-to-noise ratios (Fig. 2). When the concatenated pair is operating at 3.6 to 4.4 dB, the inner decoder runs at 0.6 to 1.4 dB. In this range, it would appear that the sparse code (302003) would work better for the inner decoder. This was borne out by simulation. Although the full code (313123) performed better as the outer code in this range, the sparse code was used for both the inner and outer codes in most of the statistics gathered. Performance of this concatenated code is compared with the single 6 by 2 decoder and a 7 by 3 decoder in Fig. 6.

The codes used in this simulation were found by a hill climbing algorithm described by J. Layland (Ref. 3). This algorithm uses an approximation of the probability of error of the code $P_e'^h$. Let I_i represent the bit sequence that is the binary expansion of the integer i, $C * I_i$ represent the coder output sequence corresponding to input I_i, and $W_H(x)$ be the Hamming weight of the sequence x. Then for a K by V code operating at a signal-to-noise ratio of E_b/N_0 define

$$P_e'^h = \sum_{\substack{i=1 \\ i \text{ odd}}}^{2^k - 1} W_H(I_i) \exp\left(\frac{-W_H(C * I_i) E_b}{V \cdot N_0}\right)$$

The algorithm starts with a given code, calculates its $P_e'^h$ and compares this with $P_e'^h$ for all codes derived by modifying one bit. If $P_e'^h$ is smaller than for all other codes in this Hamming 1-sphere, it is considered a good code. If not, the code in the sphere with smallest $P_e'^h$ is chosen and the process is repeated. The full code (313123) was the best 6 by 2 code found. The sparse code was found by using the algorithm to search for the best pair of concatenated 6 by 2 codes. It happened that both codes of the pair were the same (302003). The concatenated pair results in a 9 by 4 code (FAF43FA3C). $P_e'^h$ for this code lay between that for the best 9 by 4 code and the best 8 by 4 code. The performance of this code is shown in Fig. 6.

VI. Conclusion

Concatenation of Viterbi decoders does not appear to be useful in the present context of the planetary program. At higher signal-to-noise ratios, however, this technique might be used to produce the same performance with a small reduction in decoder complexity.

The performance difference between the best linkage scheme and direct connection is only about 0.4 dB, instead of the 2 dB that could exist if the transferred symbols were gaussian and independent. Consequently, it is conjectured that the most appropriate outer code, in any concatenation scheme involving a Viterbi algorithm inner decoder, is a high-rate algebraic block code.

References

1. Viterbi, A. J., "Decoding and Synchronization Research: Convolutional Codes: The State-Diagram Approach to Optimal Decoding and Performance Analysis for Memoryless Channels," in *Supporting Research and Advanced Development*, Space Programs Summary 37-58, Vol. III, pp. 50–55. Jet Propulsion Laboratory, Pasadena, Calif., Aug. 31, 1969.

2. Layland, J., "Information Systems: Buffer Parameters and Output Computation in an Optimum Convolutional Decoder," in *The Deep Space Network*, Space Programs Summary 37-62, Vol. II. Jet Propulsion Laboratory, Pasadena, Calif., March 31, 1970.

3. Layland, J., "Information Systems: Performance of Short Constraint Length Convolutional Codes and a Heuristic Code-Construction Algorithm," in *The Deep Space Network*, Space Programs Summary 37-64, Vol. II, pp. 44–50. Jet Propulsion Laboratory, Pasadena, Calif., Aug. 31, 1970.